



**University of
Zurich**^{UZH}

**Zurich Open Repository and
Archive**

University of Zurich
University Library
Strickhofstrasse 39
CH-8057 Zurich
www.zora.uzh.ch

Year: 2020

Neuromorphic Implementation of Spiking Relational Neural Network for Motor Control

Zhao, Jingyue ; Donati, Elisa ; Indiveri, Giacomo

Abstract: Despite the rapid development of robotic control theory, hardware motor controllers still suffer from some disadvantages: they are computationally-intensive and rely on powerful computing systems which are usually implemented using bulky and power-hungry devices. On the other hand, biological motor control systems are power-efficient, light-weight and robust. Neuromorphic engineering sheds a light on how to uncover biological control features that could lead to the design of lower power and less bulky controllers. In this paper, we present a closed-loop motor controller implemented on mixed-signal analog-digital neuromorphic hardware using a spiking neural network. The network performs PI control by encoding target, feedback and error signals using population coding. It continuously calculates the error through the network, which relates the three variables by means of feed-forward inter-population synapses. This biologically plausible and fault-tolerant strategy is ideally suited for the use of neuromorphic hardware that comprises noisy silicon neurons. Here we show how to optimize the network structure to make it robust to both noisy inputs and device mismatch. We provide experimental results showing how the controller can reach 97.1% accuracy with 75.8ms average latency.

DOI: <https://doi.org/10.1109/AICAS48895.2020.9073829>

Posted at the Zurich Open Repository and Archive, University of Zurich

ZORA URL: <https://doi.org/10.5167/uzh-189780>

Conference or Workshop Item

Accepted Version

Originally published at:

Zhao, Jingyue; Donati, Elisa; Indiveri, Giacomo (2020). Neuromorphic Implementation of Spiking Relational Neural Network for Motor Control. In: 2020 2nd IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS), Genova, Italy, 31 August 2020 - 4 September 2020. IEEE, 89-93.

DOI: <https://doi.org/10.1109/AICAS48895.2020.9073829>

Neuromorphic Implementation of Spiking Relational Neural Network for Motor Control

Jingyue Zhao*, Elisa Donati*, and Giacomo Indiveri*

*Institute of Neuroinformatics, University of Zurich and ETH Zurich, Zurich, Switzerland

Email: [jzhao, elisa, giacomo]@ini.uzh.ch

Abstract—Despite the rapid development of robotic control theory, hardware motor controllers still suffer from some disadvantages: they are computationally-intensive and rely on powerful computing systems which are usually implemented using bulky and power-hungry devices. On the other hand, biological motor control systems are power-efficient, light-weight and robust. Neuromorphic engineering sheds a light on how to uncover biological control features that could lead to the design of lower power and less bulky controllers. In this paper, we present a closed-loop motor controller implemented on mixed-signal analog-digital neuromorphic hardware using a spiking neural network. The network performs PI control by encoding target, feedback and error signals using population coding. It continuously calculates the error through the network, which relates the three variables by means of feed-forward inter-population synapses. This biologically plausible and fault-tolerant strategy is ideally suited for the use of neuromorphic hardware that comprises noisy silicon neurons. Here we show how to optimize the network structure to make it robust to both noisy inputs and device mismatch. We provide experimental results showing how the controller can reach 97.1% accuracy with 75.8ms average latency.

Index Terms—Spiking neural network, neuromorphic device, motor control, relational neural networks

I. INTRODUCTION

The control of robotic systems is a fruitful area of research. Exploitation and application of this research have been demonstrated in many domains, including industrial robot construction, domestic robotics, and biomedical prosthetics. However, compared to biological motor control systems, conventional motor controllers still have several disadvantages, especially in wearable low-power applications. The classical motor control approach focuses on producing precise movements under specific constraints and environments. The controllers are usually very computation-intensive and require powerful devices so that both the weight and power consumption do not fit in many robotic applications such as wearable devices.

A promising way to exploit and implement the principles derived from biological structures is to use neuromorphic processors for building spiking neural controllers that mimic the biological ones. Neuromorphic circuits, in fact, emulate the dynamics of biological neurons and synapses in a compact and ultra low-power technology [1]. These features make them suitable for the lightweight, always-on wearable devices, such as the motor controller. In addition, the spiking feature of the silicon neurons is a natural match for some robotic systems that need to be interfaced with the biological signals.

Neuromorphic sensors and processors are being increasingly developed and integrated into robotic systems, especially where fast, compact and power-efficient devices are required [2], [3]. However, despite their increasing deployment, a fully closed-loop neuromorphic system is still missing because motors are still being controlled by using traditional approaches. Recent works [4], [5] presented fully spiking open-loop motor controllers that used the Address Event Representation (AER) [6], [7] communication protocol implemented on Field Programmable Gate Arrays (FPGAs). In open-loop controllers, the feedback loop is absent and the controllers are less accurate in terms of result outputs and reliability. A simulated implementation of a closed-loop controller was proposed in [8] where the feedback principles were inspired by spindle populations. In this paper, the authors presented Pulse-Frequency Modulation (PFM) control rather than traditional Pulse-Width Modulation (PWM). In PFM the motor commands are encoded in the signal frequency instead of the signal width as in the PWM, therefore in PFM the spikes can be sent directly to the motor without any conversion, decreasing the latency and the power consumption. The first implementation of PFM control on a mixed-signed neuromorphic device was proposed in [9] although the system was open-loop. A different approach consists in the use of Spiking neural network (SNN) to implement traditional Proportional Integral Derivative (PID) controller. This approach aims to take the advantages of spike coding, spike information processing and the low-power consumption of their hardware implementation. Examples of spiking PID were proposed in [10] where the controller was implemented on FPGAs and in [11] on SpiN-Naker [12]. [13] implemented a neuromorphic Proportional Integral (PI) controller on mixed analog-digital device, using an SNN to perform the error calculation (i.e. subtraction). Although this work bypasses the mismatch problem of analog by only using similar neurons, there are still some drawbacks: i) the settling time is very long and, ii) the motor neuron space-coding method relies on one single neuron which can easily lead to failure.

In an effort to design a fast and fault-tolerant neuromorphic motor controller, and to improve its accuracy, we propose a closed-loop spike-based PI controller implemented on mixed-signal analog-digital neuromorphic hardware [14]. The spiking PI controller uses relational SNN to encode target, feedback and error, and to continuously calculate the error through the three variables relation encoded in the customized feed-

forward inter-population synapses. The SNN network description is presented in section II. Section III and IV show the experimental results and the effect of key network parameters on performance respectively. Finally, in section V, we summarize the results and describe possible future works.

II. MATERIALS AND METHODS

A. Neuromorphic hardware

The mixed-signal neuromorphic chip used to implement the spiking motor control network is the DYNAP-SE (Dynamic Neuromorphic Asynchronous Processors) processor [14]. The DYNAP-SE chip integrates analog circuits that emulate the behaviors of biological neurons and synapses, and digital logic circuits for communication and configuration that consist in the AER protocol. The neurons implemented on the chip are the adaptive leaky integrate-and-fire neurons that show biologically realistic behaviors. The DYNAP-SE processor consists of 4 chips, each composed of 1024 neurons with 64 programmable fan-in synapses and 4096 fan-out synapses. An SNN can be implemented on the chip by setting neuron parameters and configuring their connections. The input spikes are sent into DYNAP-SE by the configurable spike generators on peripheral FPGA using AER.

B. Neuromorphic motor controller

1) Relational neural network applied in the PI controller:

We are proposing a spiking PI controller implemented by using a threeway relational neural network [15]. Given a training dataset, this network can learn and infer any relations between several variables (e.g. $a - b = c, a = 2b = c^2$). Such variables can represent the target position that the motor must reach and the current position obtained using the motor encoder. The network is then able to calculate the error and the new motor command. The implemented SNN consists of 4 populations: 2 as inputs (target and encoder), 1 as output (error) and 1 hidden. The input/output populations represent target, feedback and error using population coding while the hidden one encodes the relationship between these variables. The connections between the input/output populations and the hidden population encode the relation and the synaptic weights can be learned using spike-based learning algorithms. The relational inference can be performed using bidirectional connections. The relation between target, feedback and motor command follows the proportional control mode where the motor command should be proportional to the error, calculated as the difference between the target and the feedback.

2) *Motor control network architecture:* Following the relational network principles, the proposed motor control network is shown in Figure 1(a), where A, B and C are 1D Winner-Take-All (WTA) [16] populations (see Figure 1(b)), which encode target, feedback and motor command respectively. H is a 2D WTA hidden population that encodes the relation $a - b = c$. The WTAs present different self-/lateral- excitations. Self-excitation is a connection from a neuron to itself and lateral-excitation is from the neuron to its neighbors along the row and column. Self-excitation is built in the input/output

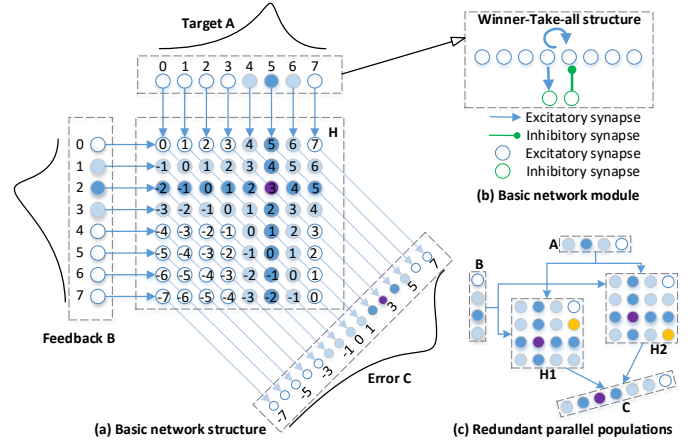


Fig. 1. Threeway relational neural network scheme.

populations and the hidden population, and lateral-excitation is used in the hidden and the output populations.

Each input population is stimulated by a group of Poisson spike generators via feed-forward, one-to-one connections. As shown in Figure 1(a), the input variables (target and feedback) are encoded in the input populations A and B using discrete magnitudes and the encoding resolution depends on the number of excitatory neurons used in the WTAs, which is configurable. In this implementation, 16 excitatory neurons and 1 inhibitory neuron are used in each input population. An input variable a is converted into spikes by setting the firing rates of the spike generators according to a Gaussian distribution with the mean set to a . In Figure 1(a), the firing rates curve of population A represents an 8-level variable $a = 5$, which mimics the biological neuron tuning curves [17]. Even if an input neuron fails here, its neighbor neuron will still encode the variable value that is closest to the correct one, which makes the network fault-tolerant.

The highest firing winner neuron in each input population (A and B) represents the encoded variable a and b respectively. The relation $a - b = c$ is encoded by feed-forward inter-population excitatory connections shown in Figure 1(a). Each neuron in A is connected to a column of neurons in H, and each neuron in B is connected to a row so that each leading diagonal in H represents a unique variable value $a - b$. The hidden matrix H is projected to the output population C where each neuron in each diagonal is connected to a single neuron in C. Ideally, in H, the most activated winner neuron is the one that receives the strongest stimulation from winner neurons in A and B. The winner neuron in H activates the corresponding winner in C. The highest firing winner in the output population C represents the subtraction result c (motor command).

The winner in C is monitored by using an exponentially-decaying spiking trace instead of a rate-based mechanism. The rate-based decoding introduces longer latency due to the unavoidable accumulation time window. In addition, since the spike counts in different time bins can vary a lot due to the non-uniform Poisson spike trains, the results are unstable and

inaccurate. The spiking trace is preferred because it is faster and more robust since it takes spike timing into account. The spiking trace value E of a neuron at time t is calculated using Equation 1:

$$E = E + 1, \text{ whenever a spike is generated}$$

$$E = E \times \exp\left(-\frac{t - t_s}{\tau}\right) \quad (1)$$

where t is the current time, t_s is the last spiking time, $\frac{1}{\tau}$ is the decay factor. Whenever a spike is generated, the trace value E increments by 1 and then it is decaying over time. The spiking trace eliminates the accumulation time of spike counting and increases the output stability because it processes historical spikes by decaying the trace value gradually instead of clearing the rate to zero.

3) *Mismatch-resistant optimizations: towards a more robust network*: Neuromorphic devices present device mismatch that needs to be considered in the network implementation. This mismatch can be considerate analog to the noise in the biological neuronal activity, and it can be challenging when present in computational tasks. In the proposed network, a slight mismatch can have a huge impact on the hidden population. Some neurons, named *outliers*, possess lower firing threshold and can fire even more than the ones that receive stronger stimulation from A and B. This behavior can mislead the winner in H resulting in the wrong winner in C.

This drawback can be solved by optimizing the hidden population. Firstly, dedicated inhibition is built by adding a 2D inhibitory neuron population with the same layout of the 2D excitatory matrix. In the inhibitory population, each neuron is excited by its corresponding excitatory neuron (named *brother neuron*) that is the one with the same location of the hidden matrix. Reversely, the inhibitory neuron helps its *brother neuron* to win the competition by inhibiting other excitatory neurons in the same row, column and minor diagonal with its *brother neuron*. This dedicated inhibition prevents the *outlier* from firing more than the expected winner. Secondly, self- and lateral-excitation are connected with a ratio of 1 : 3. Lateral-excitation must be stronger to activate the winner region where most neurons are strongly stimulated by A and B. Consequently, a single *outlier* cannot have a higher-firing winner region. In contrast, strong self-excitation may excessively activate the *outlier*. Thirdly, to reduce the effect of *outliers* in the hidden population (e.g. see neurons in yellow in Figure 1(c)), we add a redundant parallel population with the same excitatory and inhibitory neuron layout and connections. These two hidden populations independently receive the input from A and B and inject output to C. In the output population, we add lateral-excitation to further consolidate the winner region domination. In total, 1090 neurons are used in this network.

III. RESULTS

A. Time-varying neuron activities during a control experiment

Figure 2 shows the raster plot of excitatory neurons in populations A, B and C while the network is fed with

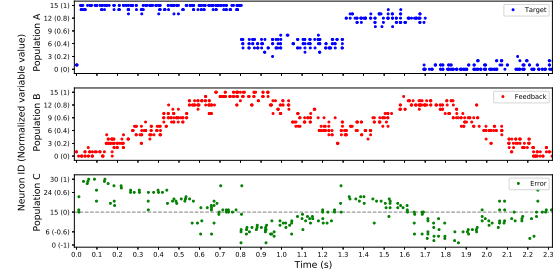


Fig. 2. Raster plot of input variable population A, B and output variable population C during a control task.

simulated target and feedback. The normalized position values encoded by populations A, B and C are variable a , b and c . Both a and b are in the range of $[0, 1]$, and c in $[-1, 1]$ all which have the resolution $\frac{1}{15}$ (given 16 neurons used in each input population). The neuron that fires most represents the encoded value of the population. The y axis of Figure 2 shows both the neuron ID and corresponding variable value. The target position a and the current one b are updated every 100 ms. In the beginning, a and b are both set to 0. A speedup process comes first during $0 \sim 0.8$ s when the target is fixed to maximum position 1 and the current position is moving gradually towards the target. During this period, the spikes of population A are mainly generated by neuron No.15 which encodes target 1, while winner neuron in B keeps changing from No.1 to No.15. The subtraction result of $a - b$ leads to a winner neuron in C that switches from No.30 to No.15. A sharp slowdown happens at 0.8s when the target position sharply changes from 1 to 0.4 but the current position is still at 1. A sharp winner change in C can be seen at this time because the calculated error suddenly changes from 0 to -0.6 . The motor direction is encoded in the sign of c : negative values represent the counterclockwise direction and positive values represent the clockwise one. With the counterclockwise rotation, current position b should decrease (as simulated here) which results in the increase of c from -0.6 to 0 between 0.8s and 1.3s. Similar speedup and slowdown processes are simulated during $1.3 \sim 1.7$ s and $1.7 \sim 2.3$ s. As shown in Figure 2, the winner change in output population C always follows the input winner change in A and B.

B. Instantaneous network state at a certain time point

Spiking trace value E is used to capture the instantaneous firing state of a neuron at a specific time point. In Figure 3 all excitatory neuron activities in A, B, C and H (sum of 2 hidden matrices) are represented by their E value given the inputs $a = 0.7, b = 0.4$. The dot-line plots on the left show the absolute E value (y axis) of each neuron (represented by its encoded variable value in x axis) in A, B and C. Under the dot-line charts, the color-coded scatter plots exhibit the relative firing magnitude of each neuron in these 3 populations. In the scatter plot of H on the right, each dot represents a neuron with a 2D coordinate. The brightest yellow neuron is

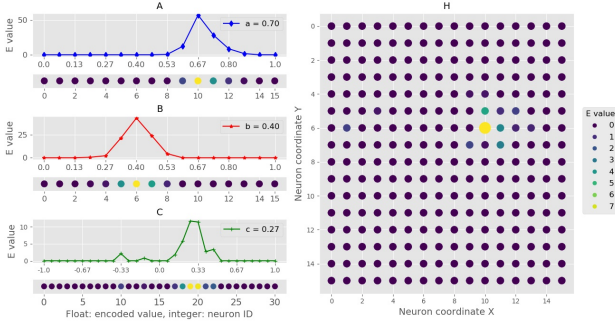


Fig. 3. Instantaneous network firing state represented by exponentially-decaying spiking trace.

TABLE I
OVERALL ACCURACY AND LATENCY.

	Decay	Inhibition	ACC	Avg. LAT (s)	Max. LAT (s)
Optimized	1	Yes	0.9759	0.1586	0.3595
		No	1.0	0.1738	0.4203
	1.5	Yes	0.9808	0.1411	0.4296
		No	0.9942	0.1301	0.267
	2.5	Yes	0.9763	0.0982	0.2321
		No	0.9668	0.1049	0.2803
Baseline	5	Yes	0.9313	0.0728	0.159
		No	0.9712	0.0758	0.1593
Baseline	5	No	0.777	0.0992	0.2281

the winner in each population. Given winner No.10 in A and No.6 in B, the winner in H should be at coordinate (10,6) which is exactly the actual winner shown in Figure 3. Winner neuron (10,6) in H activates winner No.19 in C. In the dot-line plot of C, the winner region forms a firing peak which beats the *outlier* No.10. Inside the winner region, however, we can see that neuron No.20 is also highly activated, which means the winner could switch from No.19 to its high firing neighbor No.20 during the WTA competition in all layers of the network. This is reasonable since neuron No.19 and No.20 encode 0.27 and 0.33 respectively, and the ideal result is 0.30 which is in between of these two discretized magnitudes.

C. Overall network performance

We measured accuracy and latency to test the performance of the motor control network. The latency is the time between a change in the input a/b and the generation of the correct c . The accuracy is defined as the ratio of correct decoded c given different pairs of a and b (~ 1300 samples are tested). The decoded c is considered as correct when the distance between the actual winner and the expected one in C is not larger than 2 neurons after the network converges to a stable state. As shown in Table I, the important network parameters including exponential decay factor $\frac{1}{\tau}$ and input WTA structures (with/without inhibition) are tested. The performance of a network without the 3 optimizations in the hidden population is taken as the baseline. The best accuracy the network can achieve is 100% but with 173.7ms average and 420.3ms maximum latency. Reversely, the configuration with the shortest latency (72.8ms on average and 159ms at maximum) can only reach 93.13% accuracy. The decay factor is the key parameter that affects the performance: with quicker decay, the decoded result can

switch faster to the new output when the input is changed but is less stable after the transition. Clearly, a trade-off needs to be considered. The optimal balanced solution here can achieve 97.12% accuracy and 75.8ms latency with decay $\frac{1}{\tau} = 5$, and inhibition and self-excitation used in input populations, which increases the accuracy by 25% and shortens the average latency by 24% compared to the baseline performance.

We interface the spiking motor controller on DYNAP-SE with a motor featuring an encoder. The target position is set by configuring the spike generators of population A. The current position is measured continuously by reading the motor encoder, and converted to input spikes of population B. The spiking controller calculates the motor command continuously.

IV. DISCUSSION

The trade-off between accuracy and latency is still an open issue to be further optimized. The network latency contains 3 parts: time to set the spike generators T_1 , input winner switch time T_2 and output winner switch time T_3 . T_1 costs 12 to 15 ms due to the software delay and can be shortened by FPGA. In Table I, inhibition in the input populations can slightly decrease the average latency since it can reduce T_2 from 30 ms to 23 ms which further leads to faster convergence in H and C. However, inhibition can also decrease the accuracy because it makes the output spikes from A and B less uniform (even burst if set too strong), which adds more noise into the downstream layers. T_3 is dramatically impacted by decay factor not because it can pose any physical effect on the network but due to the way we decode the output spikes. If the spiking trace decays fast, then the old winner will lose its historical advantage shortly after the new input comes. This fierce competition, however, makes the decoding suffer from instability after the transition between 2 inputs, which reduces the accuracy significantly. Self-excitation can improve accuracy by consolidating the high firing winner domination. Also, network latency can be potentially decreased by adjusting the time constant of neurons.

V. CONCLUSION

In this paper, we presented a fully spiking closed-loop PI controller implemented on mixed-signal analog-digital neuromorphic hardware. The controller can set the desired speed or position to control a single motor movements. Thanks to the features of the neuromorphic chip, this approach can be used to include learning and adaptation features, integrated into a fully spiking system using PFM. In particular, it will be interesting to test these types of controllers with robotic platforms such as the iCub robot, or in human-machine interfaces to control upper-limb prosthetic devices [18].

ACKNOWLEDGMENT

The authors would like to acknowledge Dmitrii Zendrikov, the 2019 Capocaccia Neuromorphic Workshop and all its participants for fruitful discussions. This work is supported by the Forschungskredit grant FK-18-103, the H2020 ERC project NeuroAgents (Grant No. 724295) and China Scholarship Council.

REFERENCES

- [1] E. Chicca, F. Stefanini, C. Bartolozzi, and G. Indiveri, "Neuromorphic electronic circuits for building autonomous cognitive systems," *Proceedings of the IEEE*, vol. 102, no. 9, pp. 1367–1388, 2014.
- [2] C. Bartolozzi, F. Rea, C. Clercq, D. B. Fasnacht, G. Indiveri, M. Hofstätter, and G. Metta, "Embedded neuromorphic vision for humanoid robots," in *CVPR 2011 WORKSHOPS*. IEEE, 2011, pp. 129–135.
- [3] T. Delbruck and M. Lang, "Robotic goalie with 3 ms reaction time at 4% cpu load using event-based dynamic vision sensor," *Frontiers in neuroscience*, vol. 7, p. 223, 2013.
- [4] F. Perez-Peña, A. Morgado-Estevez, A. Linares-Barranco, A. Jimenez-Fernandez, F. Gomez-Rodriguez, G. Jimenez-Moreno, and J. Lopez-Coronado, "Neuro-inspired spike-based motion: from dynamic vision sensor to robot motor open-loop control through spike-vite," *Sensors*, vol. 13, no. 11, pp. 15 805–15 832, 2013.
- [5] F. Perez-Peña, A. Linares-Barranco, and E. Chicca, "An approach to motor control for spike-based neuromorphic robotics," in *2014 IEEE Biomedical Circuits and Systems Conference (BioCAS) Proceedings*. IEEE, 2014, pp. 528–531.
- [6] S. R. Deiss, R. J. Douglas, A. M. Whatley *et al.*, "A pulse-coded communications infrastructure for neuromorphic systems," *Pulsed neural networks*, pp. 157–178, 1999.
- [7] K. A. Boahen, "Point-to-point connectivity between neuromorphic chips using address events," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 47, no. 5, pp. 416–434, 2000.
- [8] Perez-Peña, J. A. Leñero-Bardallo, A. Linares-Barranco, and E. Chicca, "Towards bioinspired close-loop local motor control: A simulated approach supporting neuromorphic implementations," in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2017, pp. 1–4.
- [9] E. Donati, F. Perez-Peña, C. Bartolozzi, G. Indiveri, and E. Chicca, "Open-loop neuromorphic controller implemented on vlsi devices," in *2018 7th IEEE International Conference on Biomedical Robotics and Biomechatronics (Biorob)*. IEEE, 2018, pp. 827–832.
- [10] A. Jimenez-Fernandez, G. Jimenez-Moreno, A. Linares-Barranco, M. J. Dominguez-Morales, R. Paz-Vicente, and A. Civit-Balcells, "A neuro-inspired spike-based pid motor controller for multi-motor robots with low cost fpgas," *Sensors*, vol. 12, no. 4, pp. 3831–3856, 2012.
- [11] F. Galluppi, C. Denk, M. C. Meiner, T. C. Stewart, L. A. Plana, C. Eliasmith, S. Furber, and J. Conrad, "Event-based neural computing on an autonomous mobile platform," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 2862–2867.
- [12] S. B. Furber, F. Galluppi, S. Temple, and L. A. Plana, "The spinnaker project," *Proceedings of the IEEE*, vol. 102, no. 5, pp. 652–665, 2014.
- [13] S. Glatz, J. N. Martel, R. Kreiser, N. Qiao, and Y. Sandamirskaya, "Adaptive motor control and learning in a spiking neural network realised on a mixed-signal neuromorphic processor," *arXiv preprint arXiv:1810.10801*, 2018.
- [14] S. Moradi, N. Qiao, F. Stefanini, and G. Indiveri, "A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (dynaps)," *IEEE transactions on biomedical circuits and systems*, vol. 12, no. 1, pp. 106–122, 2017.
- [15] P. U. Diehl and M. Cook, "Learning and inferring relations in cortical networks," *arXiv preprint arXiv:1608.08267*, 2016.
- [16] T. Fukai and S. Tanaka, "A simple neural network exhibiting selective activation of neuronal ensembles: from winner-take-all to winners-share-all," *Neural computation*, vol. 9, no. 1, pp. 77–97, 1997.
- [17] A. Schoups, R. Vogels, N. Qian, and G. Orban, "Practising orientation identification improves orientation coding in v1 neurons," *Nature*, vol. 412, no. 6846, p. 549, 2001.
- [18] E. Donati, M. Payvand, N. Risi, R. B. Krause, and G. Indiveri, "Discrimination of emg signals using a neuromorphic implementation of a spiking neural network," *IEEE transactions on biomedical circuits and systems*, 2019.